Problem Statement:

Customer churn is when a company’s customers stop doing business with that company. Businesses are very keen on measuring churn because keeping an existing customer is far less expensive than acquiring a new customer. New business involves working leads through a sales funnel, using marketing and sales budgets to gain additional customers. Existing customers will often have a higher volume of service consumption and can generate additional customer referrals.

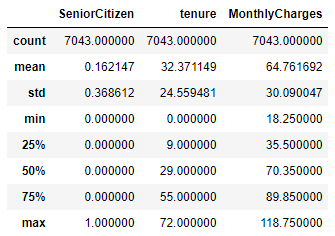
Customer retention can be achieved with good customer service and products. But the most effective way for a company to prevent attrition of customers is to truly know them. The vast volumes of data collected about customers can be used to build churn prediction models. Knowing who is most likely to defect means that a company can prioritise focused marketing efforts on that subset of their customer base.

Preventing customer churn is critically important to the telecommunications sector, as the barriers to entry for switching services are so low.

You will examine customer data from IBM Sample Data Sets with the aim of building and comparing several customer churn prediction models.

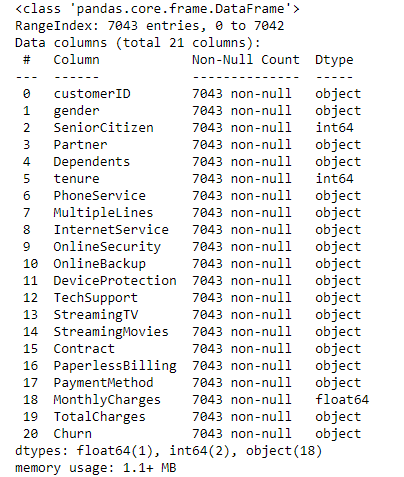
 Data Analysis:

A quick describe method reveals that the telecom customers are staying on average for 32 months and are paying $64 per month. However, this could potentially be because different customers have different contracts.



We can presume that the dataset contains several numerical and categorical columns providing various information on the customer transactions.

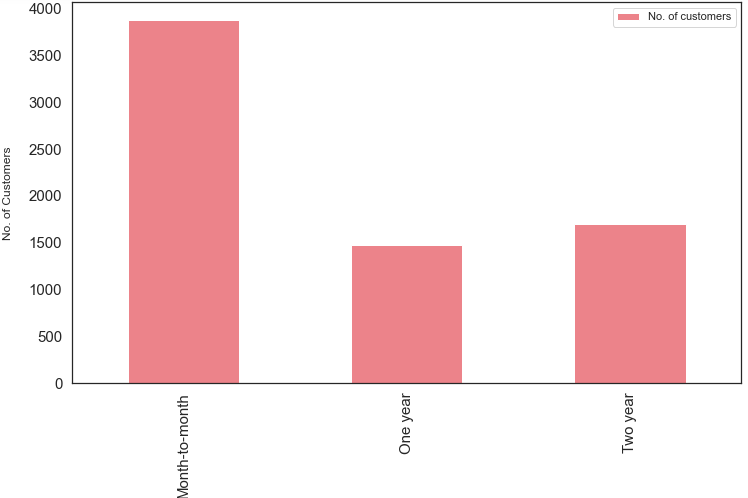
Keeping an eye onto the missing values in a dataset. The missing values could mess up model building and accuracy. Hence we need to take care of missing values (if any) before we compare and select a model. The dataset contains 7043 rows and 21 columns and there seem to be no missing values in the dataset.

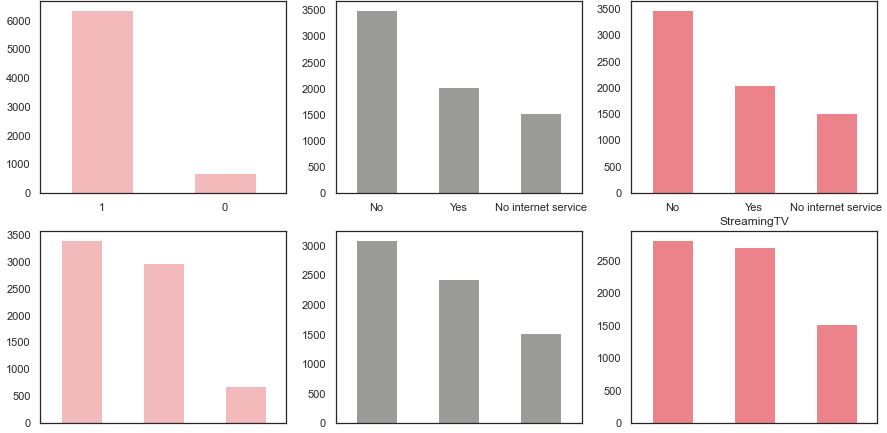


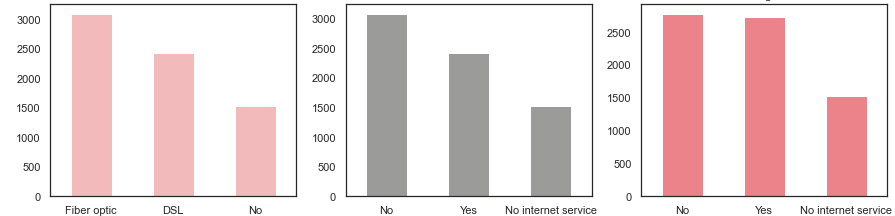
Let’s look at the distribution of churn values. This is quite a simple yet crucial step to see if the dataset upholds any class imbalance issues. As you can see above, the data set is imbalanced with a high proportion of active customers compared to their churned counterparts.



Most of the customers seem to have a prepaid connection with the telecom company. On the other hand, there are a more or less equal proportion of customers in the 1-year and 2-year contracts.







Most of the customers have phone service out of which almost half of the customers have multiple lines. 3/4th of the customers have opted for internet service via Fiber Optic and DSL connections with almost half of the internet users subscribing to streaming TV and movies. Customers who have availed Online Backup, Device Protection, Technical Support and Online Security features are a minority.

EDA Concluding Remarks:

The dataset does not have any missing or erroneous data values. Strongest positive correlation with the target features is Monthly Charges and Age whilst negative correlation is with Partner, Dependents and Tenure. The dataset is imbalanced with the majority of customers being active. There is multi-collinearity between Monthly Charges and Total Charges. Dropping Total Charges have decreased the VIF values considerably. Most of the customers in the dataset are younger people. There are a lot of new customers in the organization (less than 10 months old) followed by a loyal customer base that’s above 70 months old. Most of the customers seem to have phone service with Monthly charges spanning between 18 to118 per customer. Customers with a month-to-month connection have a very high probability to churn that too if they have subscribed to pay via electronic checks.

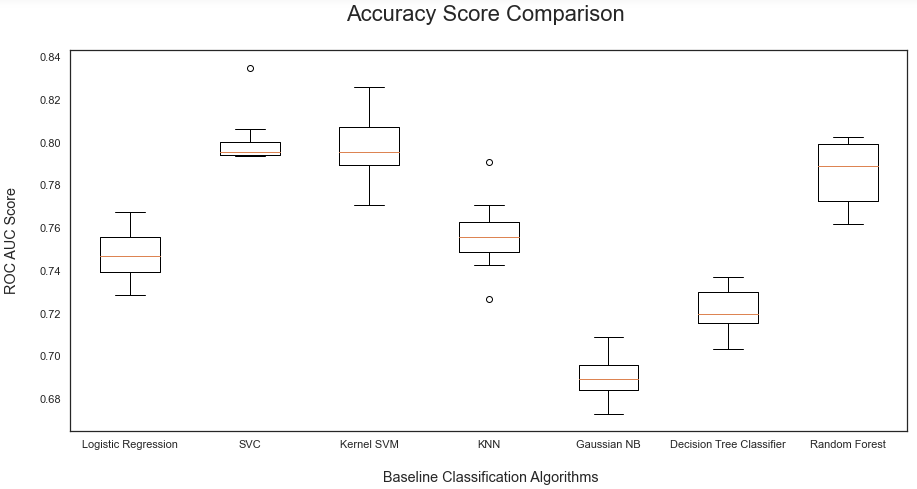
Pre-processing Pipeline:

Any categorical variable that has more than two unique values have been dealt with Label Encoding and one-hot Encoding using get\_dummies method in pandas.

Conduct Feature Scaling: It’s quite important to normalize the variables before conducting any machine learning (classification) algorithms so that all the training and test variables are scaled within a range of 0 to 1.

Let’s model each classification algorithm over the training dataset and evaluate their accuracy and standard deviation scores.

Classification Accuracy is one of the most common classification evaluation metrics to compare baseline algorithms as its the number of correct predictions made as a ratio of total predictions. However, it's not the ideal metric when we have class imbalance issue. Hence, let us sort the results based on the ‘Mean AUC’ value which is nothing but the model’s ability to discriminate between positive and negative classes.

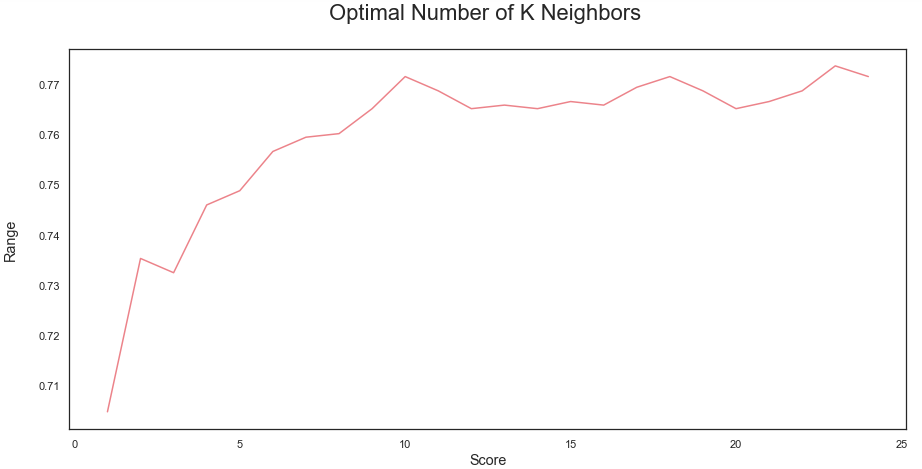


Building Machine Learning Models:

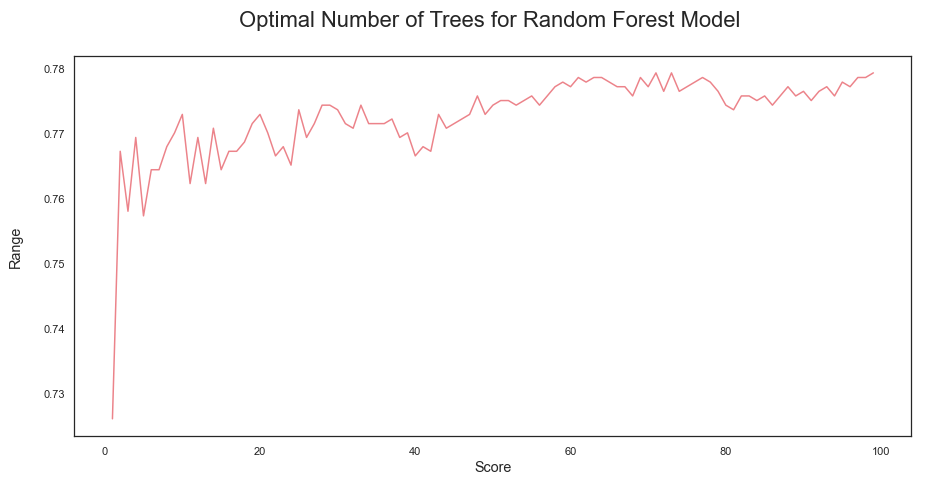
Get the right parameters for the baseline models: Before doing the second iteration, let’s optimize the parameters and finalize the evaluation metrics for model selection.

Identify the optimal number of K neighbours for KNN Model: In the first iteration, we assumed that K = 3, but in reality, we don’t know what is the optimal K value that gives maximum accuracy for the chosen training dataset. Therefore, let us write a for loop that iterates 20 to 30 times and gives the accuracy at each iteration so as to figure out the optimal number of K neighbours for the KNN Model.

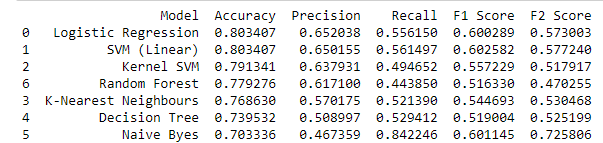
As we can see from the iterations, if we use K = 22, then we will get the maximum score of 78%.



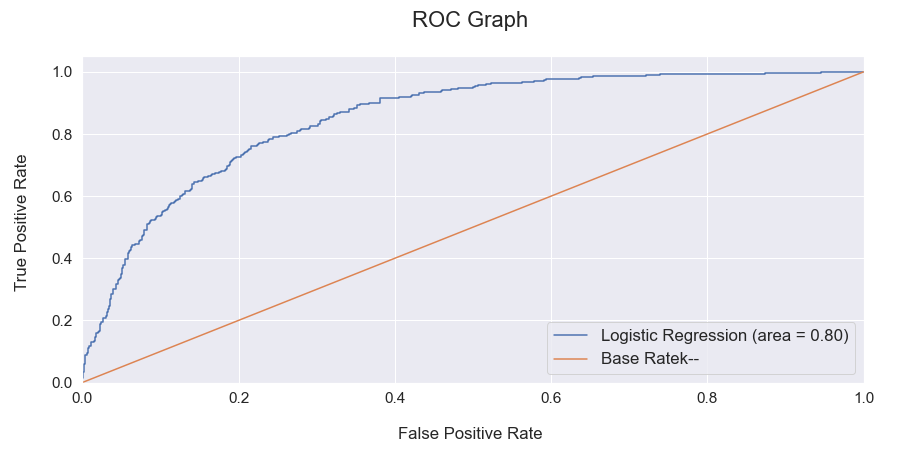
Identify the optimal number of trees for Random Forest Model: Quite similar to the iterations in the KNN model, here we are trying to find the optimal number of decision trees to compose the best random forest. As we could see from the iterations below, the random forest model would attain the highest accuracy score when its n\_estimators = 72.



From the 2nd iteration, we can definitely conclude that logistic regression is an optimal model of choice for the given dataset as it has relatively the highest combination of precision, recall and F2 scores; giving most number of correct positive predictions while minimizing the false negatives. Hence, let's try to use Logistic Regression and evaluate its performance in the forthcoming sections.



Evaluate the model using ROC Graph: It’s good to re-evaluate the model using ROC Graph. ROC Graph shows us the capability of a model to distinguish between the classes based on the AUC Mean score. The orange line represents the ROC curve of a random classifier while a good classifier tries to remain as far away from that line as possible. As shown in the graph below, the fine-tuned Logistic Regression model showcased a higher AUC score.



Predict Feature Importance: Logistic Regression allows us to determine the key features that have significance in predicting the target attribute (“Churn” in this project).

The logistic regression model predicts that the churn rate would increase positively with month to month contract, optic fibre internet service, electronic checks, absence of payment security and tech support.

On the other hand, the model predicts a negative correlation with churn if any customer has subscribed to online security, one-year contract or if they have opted for mailed checks as their payment medium.

Model improvement basically involves choosing the best parameters for the machine learning model that we have come up with. There are two types of parameters in any machine learning model — the first type are the kind of parameters that the model learns; the optimal values automatically found by running the model. The second type of parameters is the ones that user get to choose while running the model. Such parameters are called the hyper-parameters; a set of configurable values external to a model that cannot be determined by the data, and that we are trying to optimize through Parameter Tuning techniques like Random Search or Grid Search.

Concluding Remarks:

Unpredictability and risk are the close companions of any predictive models. Therefore in the real world, its always a good practice to build a propensity score besides an absolute predicted outcome. Instead of just retrieving a binary estimated target outcome (0 or 1), every ‘Customer ID’ could get an additional layer of propensity score highlighting their percentage of probability to take the target action.

So, in a nutshell, we made use of a customer churn dataset from Kaggle to build a machine learning classifier that predicts the propensity of any customer to churn in months to come with a reasonable accuracy score of 76% to 84%.